

## Supplementary material

### A Shotgun read

```
>4_3/1 reference=gi|45439865| position=3821587-3821680 strand=-1 description="Yersinia pestis biovar Microtus str. 91001"
```

↑  
Read ID, including library no. (left) and mate pair no. (right)

↑  
ID of the reference sequence the read was taken from

↑  
Start and stop position on the reference

↑  
Reverse strand

↑  
Description of the reference sequence

### B Amplicon read

```
>4_3/1 reference=gi|45439865| amplicon=12936-13436 position=1-102 strand=+1 mid=ACGT description...
```

↑  
Position of the PCR amplicon on the reference

↑  
Read position on the amplicon

↑  
Forward strand

↑  
Sequence of the multiplex identifier barcode

### C Amplicon read with errors

```
>4_3/1 reference=gi|45439865|,gi|31791177| amplicon=12936-13436,1470781-1471280 position=1-92 strand=+1 errors=24%A,55+TT,68- mid...
```

↑  
Multiple references or amplicons indicate a chimera

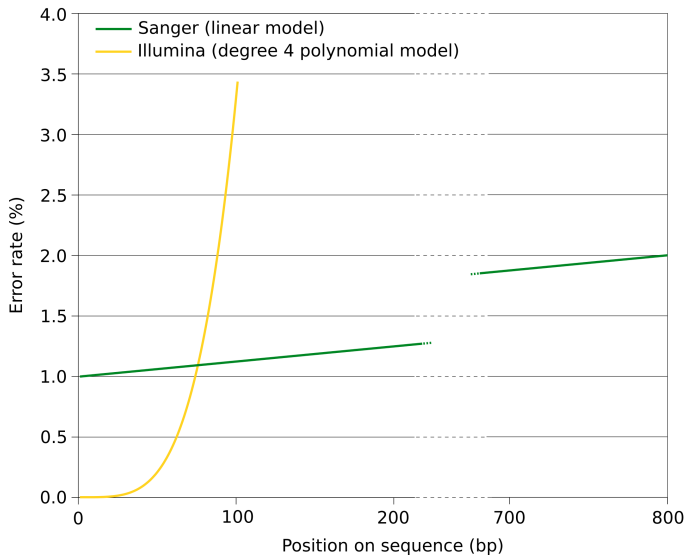
↑  
Substitution of the 24th amplicon or chimera residue by a T

↑  
Insertion of TT after the 55th residue

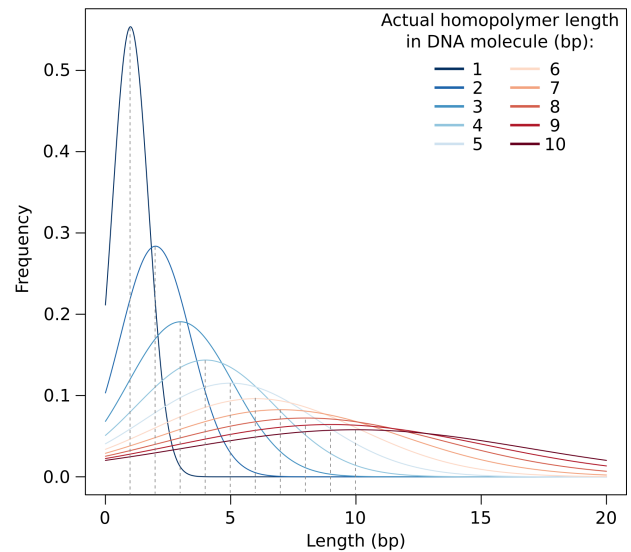
↑  
Deletion of the 68th residue

Supplementary Figure 1: Grinder read traceability. Example of the structured FASTA description for A) a simulated shotgun read, B) a simulated amplicon read, and C) a simulated chimeric amplicon read containing sequencing errors.

### A Sanger and Illumina position-specific error model



### B 454 homopolymer error model



Supplementary Figure 2: Error models implemented in Grinder: A) Sanger and Illumina position-specific model, B) 454 homopolymer error model.

Galaxy / ACE Analyze Data Workflow Shared Data Admin Help User

Tools Options

grinder

NGS: Simulation

- Grinder sequence simulator

Workflows

---

**Grinder**

Specify: Built-in file

Reference sequences: NCBI nt  
Galaxy built-in FASTA file

Specify: Total number of reads

Number of reads or mate pairs to create: 100  
Number of shotgun reads or mate pairs to generate. Do not specify this if you specify the coverage

Sequence length distribution: 100 normal 10  
Desired sequence length distribution specified as: average length, distribution ('uniform' or 'normal') and standard deviation Only the first element is required. Examples: 1/ All sequences exactly 250 bp long: 250 2/ Uniform distribution around 100+-10 bp: 100 uniform 10 3/ Read normally distributed with an average of 800 and a standard deviation of 100 bp: 800 normal 100

Insert size distribution: 0  
Create shotgun paired end reads (mate pairs) spanning the given insert length (the reads are interior to the insert): 0 : off, or: insert size distribution in bp, in the same format as the read length distribution (a typical value is 2,500 bp) Two distinct reads are generated whether or not the mate pair overlaps. Default: insert\_dist.default

Characters to exclude: "  
Do not create reads containing any of the specified characters (e.g.,

---

History Options

Grinder work

3: Grinder on data 1 23.1 Kb format: fasta, database: ?

```
>1 source=gi|20428553|ref|NC_003765.1| pc
TTGGTGAGGCATTTCATCTACTCCAGCATAAAGTCAAAGTGA
GACAAGTCGAAGTTATTCTCATGCTCCCTCGTTAAGTAA
>2 source=gi|298209958|ref|NC_014243.1| p
GATTCGGATGCCATCCATCATATATTCTAGTTAAAACTCT
TTGATGTATCCATCCGTATTGTTAATTCACCCCTGCCGA
```

2: Grinder on data 1

1: viral\_db.fna

Supplementary Figure 3: Screenshot of the Grinder Galaxy GUI

```

synopsis.pl - gedit
File Edit View Search Tools Documents Help
Open Save Undo
synopsis.pl x
use Grinder;

# Set up a new factory (see the OPTIONS section for a complete list of parameters)
my $factory = Grinder->new( -reference_file => 'genomes.fna' );

# Process all shotgun libraries requested
while ( my $struct = $factory->next_lib ) {

    # The ID and abundance of the 3rd most abundant genome in this community
    my $id = $struct->{ids}->[2];
    my $ab = $struct->{abs}->[2];

    # Create shotgun reads
    while ( my $read = $factory->next_read ) {

        # The read is a Bioperl sequence object with these properties:
        my $read_id = $read->id; # read ID given by Grinder
        my $read_seq = $read->seq; # nucleotide sequence
        my $read_mid = $read->mid; # MID or tag attached to the read
        my $read_errors = $read->errors; # errors that the read contains

        # Where was the read taken from?
        my $ref_id = $read->reference->id; # ID of the reference sequence
        my $ref_start = $read->start; # start of the read on the reference
        my $ref_end = $read->end; # end of the read on the reference
        my $ref_strand = $read->strand; # strand of the reference
    }
}

# Similarly, for shotgun mate pairs
my $factory = Grinder->new( -reference_file => 'genomes.fna',
                          -insert_dist => 250 );
while ( $factory->next_lib ) {
    while ( my $read = $factory->next_read ) {
        # The first read is the first mate of the mate pair
        # The second read is the second mate of the mate pair
        # The third read is the first mate of the next mate pair
        # ...
    }
}

# To generate an amplicon library
my $factory = Grinder->new( -reference_file => 'genomes.fna',
                          -forward_reverse => '16Sgenes.fna',
                          -length_bias => 0,
                          -unidirectional => 1 );
while ( $factory->next_lib ) {
    while ( my $read = $factory->next_read ) {
        # ...
    }
}
Perl Tab Width: 8 Ln 44, Col 63 INS

```

Supplementary Figure 4: Overview of the methods provided by the Grinder object-oriented API.

Supplementary Dataset 1: Twelve 16S rRNA amplicon libraries simulated using Grinder (ten with homopolymer errors and two without).